

HP customer perspective: how IT quality managers respond to SOA change
White paper



Table of contents

Executive summary: the price of agility2
SOA: the quality challenge2
Volume and complexity2
New technologies3
Generality4
Meeting the challenge4
Automate lifecycle and test management4
Allocate testing effort based on risk5
Implement SOA-aware tools6
Implement all three pillars of a comprehensive test program7
Turning challenges to opportunity8
Learn more8

The quality organization should view every service as a “mini-application” and consider all the ways it might be used.

Executive summary: the price of agility

Service-oriented architecture (SOA) promises to shorten development cycles, increase re-use, and make enterprises better able to respond to business change. And it does. But SOA requires IT professionals at all levels to rethink what they do and how they do it. Dr. Ferhan Kilical, now test manager at Northrop Grumman Mission Systems, has been managing the testing of large-scale, mission-critical systems built on SOA for over five years. “SOA works!” she says. “But it changes everything for the quality organization.”

Managing IT quality has always been a hard job. There are too many applications and not enough time and resources. Test effort must be allocated based on the risk presented by each module. In SOA, re-usable services can amplify risk—a failure in a critical service can ripple through all the applications and other services that access it. And SOA means more things to test and more lifecycles to manage. Traditional test tools—designed to test application transactions—provide little help in testing services with broad functionality and no user interface. Dr. Kilical uses HP Quality Center software and HP Performance Center software to implement a risk-based testing program for SOA that uses a lifecycle approach to track quality from requirements through testing to release. This paper describes the challenges Dr. Kilical and other IT quality managers face when applying traditional test tools and processes to SOA, and it shows how HP has adapted HP Quality Center and HP Performance Center software to help IT meet the new needs of SOA.

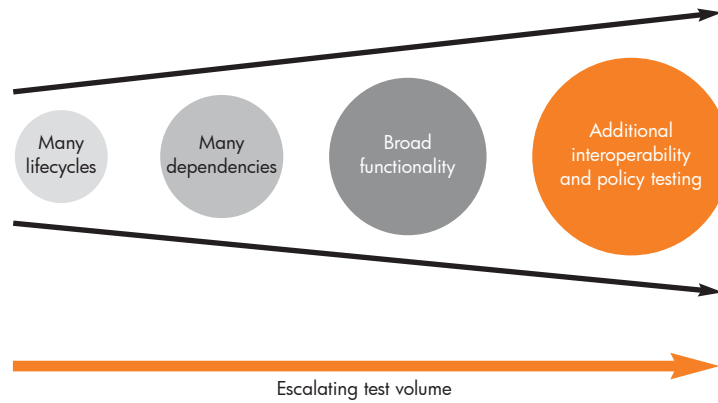
SOA: the quality challenge

Top-notch test teams plan the test effort to be applied to each facet of the application, use automated tools to perform tests efficiently, and analyze results to make release decisions. But SOA—inherently distributed, loosely coupled, asynchronous, and heterogeneous—brings new challenges to established processes. Dr. Kilical, a Six Sigma Green Belt, speaks in the breathless flurry of a harried IT manager juggling dozens of projects. She describes a complex environment built on technologies often new to the test organization and presenting a level of generality that makes it difficult to know where to apply the most test effort. “There are many more lifecycles to manage in SOA. We have to test each service as it’s developed, then we have to test the integrations between services and the applications that use them. And some services have no GUI, so our traditional test tools didn’t help at all.”

Volume and complexity

It is the complexity that presents much of the challenge. SOA requires the development organization to build the services, test them, and then build applications that access them. And when services change—and they change often—regression testing must validate that the other services and applications that access them still work. As a result, the number of independent lifecycles to be managed grows exponentially, especially when the organization adopts agile development methods that roll out functionality in iterations. Test teams need fast, efficient test processes to handle the volume of testing generated in a SOA. And the test organization needs a way to keep track of which applications and services use which other services so regression testing can be done when needed.

Figure 1. SOA increases test volume and complexity.



SOA dramatically increases the amount of testing required.

And there is more to be tested—the sheer volume of testing required is dramatically increased. In each test cycle, testing must confirm not only that services and applications work as specified but that they conform to functionality, performance, and security policies established by SOA governance. Policies may dictate certain mandatory methods for objects, or specify how they must handle exceptions. Security policies are carefully specified for every service. Conformance to all of these policies must be verified.

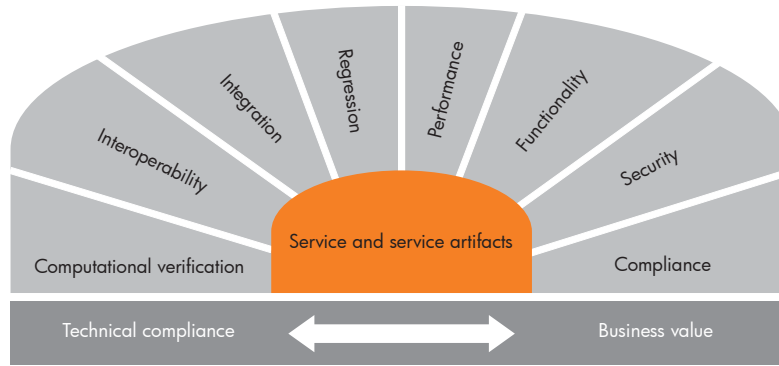
Services and applications may be built by different development teams using different development environments and toolkits, so interoperability testing is critical. The Web Services Interoperability Organization (WS-I) has established profiles to help assure interoperability. Those tests must be run before the service can claim WS-I compliance. “We have teams using Java™ and other development tools. And some services are provided by third parties who may use something different still. It’s supposed to work, but we have to test it to be sure.” Finally, because SOA is loosely coupled, there are more boundary conditions to test. All this means more lifecycles and more testing per lifecycle.

New technologies

SOA not only brings new methodologies, it brings new technologies. Testers who previously worked with application user interfaces must be able to understand, test, and analyze service functionality in terms of their interface definitions alone—WSDL and XML. Web services have no user interface—and it’s certainly not feasible to develop one for test purposes—so testers must interact with the interface definition to structure and initiate tests and analyze results. Services rely on new protocols and facilities like SOAP and UDDI. This imposes new training requirements on both developers and testers. Dr. Kilical says, “Testers must get a little smarter with SOA, but good tools really help.”

One area where new technologies are introduced is security. “Traditional IT security is based on a person-to-machine security model, but SOA requires a machine-to-machine model,” Dr. Kilical points out. Authentication is usually accomplished with security certificates rather than passwords; this adds complexity, and it adds to the technologies that the quality organization must understand and manage. On top of it all, the technology is new and standards are still evolving. Work by standards bodies can obsolete tools and learning almost overnight.

Figure 2. The SOA test spectrum—balancing technical compliance and business value



IT organizations implementing SOA must implement new best practices and tools.

Generality

Traditional applications need only meet the identified functionality requirements of the users. But services deployed in a SOA must be designed to meet the needs of users of many applications, most of which have not been developed yet. As a result, services usually offer very broad interfaces with inputs, outputs, and methods not yet in general use. The quality organization should view the service as a “mini-application” and consider all the ways it might be used. But since test time and test resources are limited, testers must decide where to best invest the resources they have. Analysts and developers often take the approach that if certain data is available, the service interface should provide access to it. But testers must develop the business knowledge to understand which ones are most critical and will most affect the business if defects go undetected.

Meeting the challenge

Dr. Kilical has built a test program for SOA based on several key principles and supported by HP Quality Center and HP Performance Center software. “In my job prior to Northrop Grumman, we did not have HP Service Test software, and it was really painful to test SOA. The services did not have a GUI, and we had to search through never-ending log files to understand the failures. HP had updated their automated software quality systems to provide the new functionality we needed. It makes it a lot easier to test SOA applications with the new tool.” Dr. Kilical believes IT quality organizations implementing SOA must use best practices and tools designed to support quality management processes and optimized for SOA.

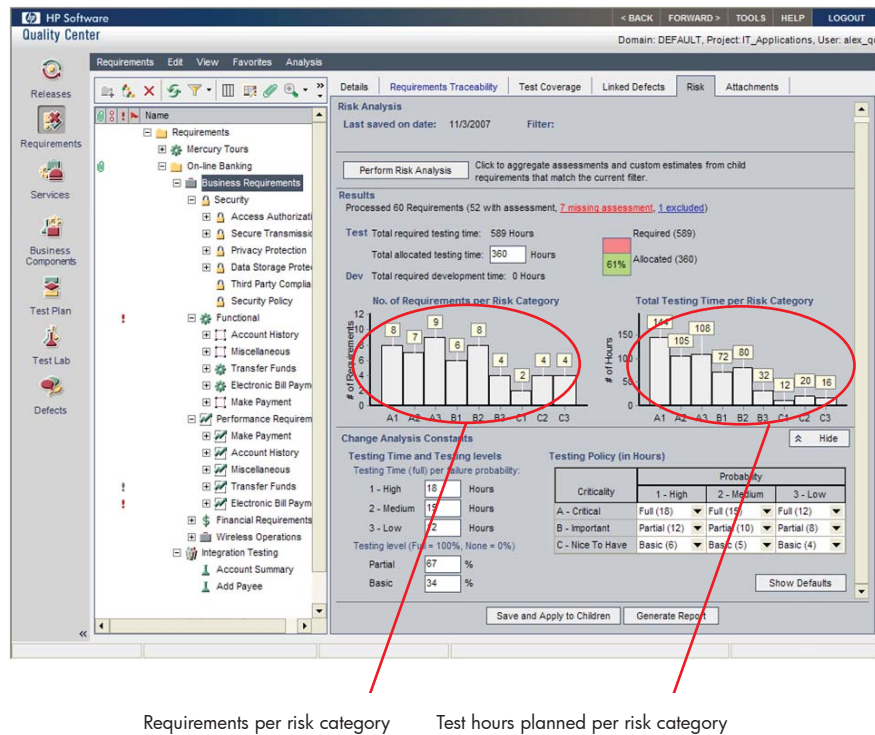
Automate lifecycle and test management

“We couldn’t do it without automation,” she says. “There are just too many lifecycles to manage and too many tests to plan, perform and analyze. We have to plan a full battery of tests—simulation, unit testing, data management, functional and regression testing, interoperability, governance policies, vulnerability and security, and performance and load testing.”

During automated test planning, test resources—people and equipment—are allocated to requirements based on risk and priority. As with other companies, the Northrop Grumman Mission Systems team maintains a master test plan with a schedule that includes the new services and applications that need to be tested, the services expected to change and need retesting, the kinds of tests that must be performed, and the expected number of test cycles. The plan provides a central planning point where managers can estimate resource needs, view testing and defect status, and identify testing bottlenecks in time to redeploy testers and lab resources when necessary. When changes to the overall project schedule are jeopardized by lack of test resources, test managers are not only aware but have the data needed to show senior managers why there is a problem and what must be done.

Having an automated planning process enables the quality organization to begin testing a service as early as possible. “Developers know they have to get a service in the system to get it tested,” Dr. Kilical notes. This is important because the earlier defects are found, the easier and less expensive they are to fix. And once in the system, services can be managed throughout their lifecycle. “When deployed services need retesting because of changes, we already have the data we need to do it quickly.”

Figure 3. Managing test coverage by risk category



Requirements per risk category Test hours planned per risk category

HP Quality Center manages quality across the entire service lifecycle so testing can begin early and continue through service obsolescence. It shows test managers where resources are needed and helps them respond to the change that is inevitable in SOA. It reflects the current status of requirements, tests, and defects; and it tracks quality data for the lifetime of a service or application.

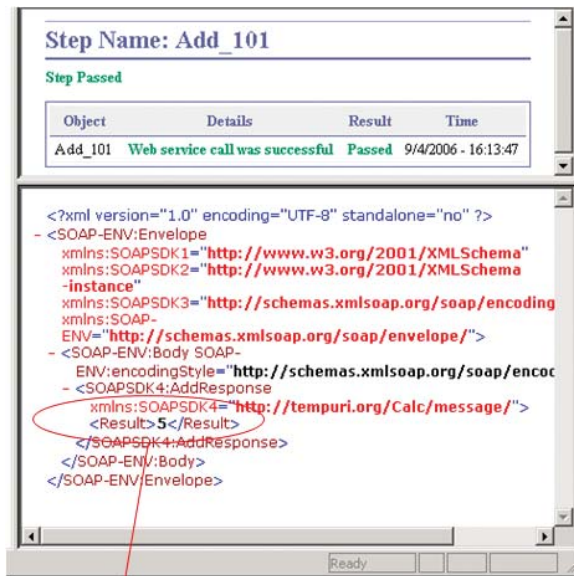
Allocate testing effort based on risk

“There is never enough time and never enough resources,” Dr. Kilical contends. To use resources effectively, you must allocate them based on the risk presented by each service and each interface function to be tested. Risk is a product of the business impact of a failure and the probability a failure will occur. When services are re-used by other services and many applications, risk increases substantially. Some simple

rules can be applied—methods that change data are generally more risky than methods that only retrieve data—but Dr. Kilical relies on a business understanding of how services are used and on close collaboration with analysts and developers to estimate how many applications and other services will eventually use a service or function.

HP Quality Center helps by providing a structured way of quantifying risk and by comparing planned test effort to risk. “HP Quality Center aids senior managers [to] visualize and quantify the level of effort required in SOA testing. It also helps them to see the most effective use of the resources we have,” according to Dr. Kilical.

Figure 4. HP Service Test is SOA-aware.



HP Service Test results in XML

Test systems must be able to connect to SOA repositories.

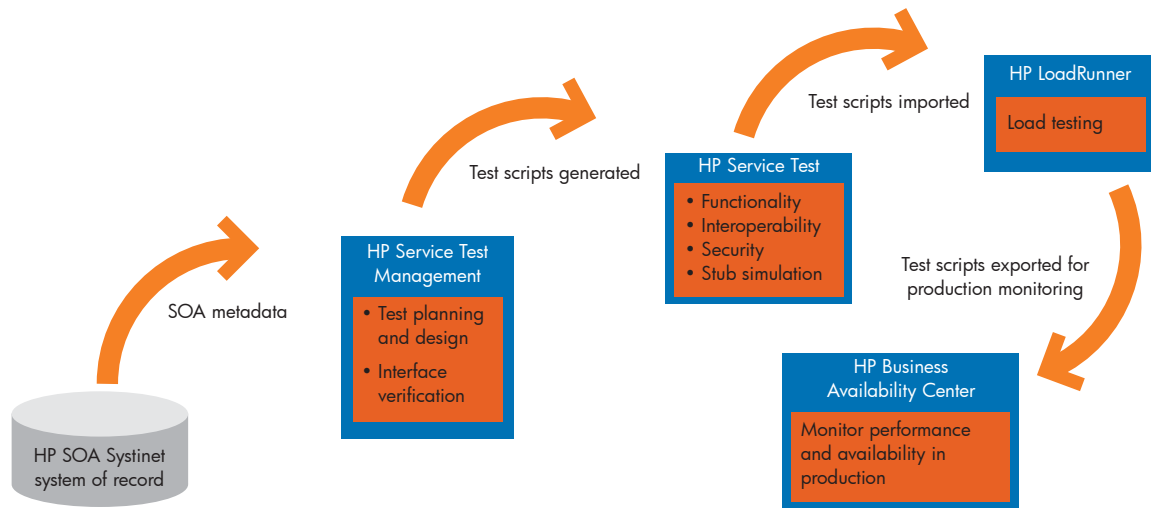
Implement SOA-aware tools

Test automation systems must understand SOA and Web services. Test software must support Web services facilities and protocols like SOAP and UDDI. And since services have no user interface, traditional testing techniques and tools that generate test scripts based on GUI interaction just don't work. Test tools must be able to digest WSDL interface definitions and enable testers to design tests based on them. Results must point back to these same artifacts so testers and developers can see exactly which methods and outputs have failed. Test systems must be able to connect to SOA repositories and systems of record to access interface definitions and other artifacts associated with the service. And it is these repositories that keep track of who is using each service—critical information when services change and regression testing must be done.

HP Service Test software—a plug-in to HP Quality Center—bridges the gap between traditional testing techniques and SOA. HP Service Test accesses service artifacts through a SOA system of record like HP SOA Systinet software using UDDI or other sharing mechanisms. It imports WSDL and validates the XML contained in it. Then it enables testers to plan, design, execute, and analyze tests of the service based on the WSDL. No user interface is needed. A common problem is that some services require access to others before they are completely developed. HP Service Test enables developers or testers to create service simulations that provide parameterized responses to requests from applications or other services. This capability enables full testing of modules before dependencies are implemented, and that accelerates testing.

HP Service Test supports a broad set of protocols including SOAP, Web services over HTTP, Web services over Java Message Service, and IBM WebSphere MQ; and it works with the most popular Web services toolkits like the Apache Axis framework.

Figure 5. SOA testing with HP software



Using functional scripts in performance testing is “a huge timesaver.”

Implement all three pillars of a comprehensive test program

Dr. Kilical says: “Any test program must test functionality, performance and security. Each is critical to the success of applications in a SOA.” Functionality is specified by the interface definition. Each method and each input and output must be tested according to the risk it represents. But it is also essential to characterize the scalability of a service before it is published for use by many applications. Performance bottlenecks in SOA services—like functionality failures—ripple through all of the other services and applications that rely on them. When services of outside service providers are used, service level agreements may specify response and service levels. These must be tested and verified to ensure the service will meet the needs foreseen for it (and paid for). While security testing is critical in most applications, Dr. Kilical finds it takes on an even larger role in SOA because vulnerabilities may be exposed to and exploited by a much broader user base.

HP Quality Center and HP Performance Center software enable testing organizations to deliver a comprehensive testing effort with the resources available. HP Service Test can wring out the functionality and security of a service, and HP LoadRunner software allows testers to simulate any load when testing applications or services and measure the response and throughput. With the SOA plug-in for HP LoadRunner, scripts generated by HP Service Test can be imported into HP LoadRunner. “Services can be tested under any load,” Dr. Kilical reports, “and without having to create new scripts. It’s a huge timesaver.”

Figure 6. A comprehensive test program for SOA

Lifecycle automation	Risk-based testing	SOA-aware tools	3-pillar program
<ul style="list-style-type: none">• Improve efficiency• Increase visibility	<ul style="list-style-type: none">• Optimize test resources• Align with business priorities	<ul style="list-style-type: none">• Handle SOA specifics• Reduce test cycle time	<ul style="list-style-type: none">• Functionality• Performance• Security

Turning challenges to opportunity

SOA presents great challenges for test teams, but it is also an opportunity for the quality organization to shine. Dr. Kilical says, "Our test teams know more about the business and contribute more business value than before SOA. Since we enter the process so early, test becomes a meeting ground for analysts, architects, developers, database administrators, security specialists and others." Quality management not only verifies that user requirements are met, it confirms that the governance framework prescribed by the organization is actually implemented. This confirmation is critical to the success of any SOA. It builds trust that shared services are secure, that they operate according to the artifacts presented with them, that they can support the loads expected, and that change will not disrupt applications that rely on the shared services.

Achieving quality with SOA requires quality management teams to implement processes suited to SOA. And it requires them to support these processes with tools designed for SOA—tools like HP Quality Center and HP Performance Center software. More than 60 percent of the organizations that use an automated software quality solution use HP software. When they are ready for the move to SOA, they find that HP is already there.

Learn more

To learn more about HP solutions for SOA and service-oriented quality, visit www.hp.com/go/soa or contact your HP representative.

To learn more, visit www.hp.com/go/soa

© 2008 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Java is a U.S. trademark of Sun Microsystems, Inc.

4AA2-0628ENW, June 2008



Technology for better business outcomes